

킵오프

- 테이 이어몬 그리고 깡뚜기 오톨의 자바스크립트는 왜 그따구인가 스테디.

취지

- js 개발자면 더글라스 크락포트가 쓴 책 한권쯤 읽어보자!(자바스크립트 개발에 참여, json, jsLint 만들었음), 2008년에 쓴 나비책은 그 당시 js 개발자들의 바이블이었다고 함.
- js의 문법이나 내부 구조를 다루는 좋은 책은 많음(모자스,코어자스) 근데 이 책은 js 의 구현체를 깊게 다루지는 않는것같음.
- 그보다는 좀더 제너럴한 프로그래밍 언어론에 대해 다루고, 자바스크립트가 이렇게 된 이유를 (그리고 나아가서는 대안)을 제시해주는 책인듯.
- 이 책을 관통하는 취지는 자바스크립트의 좋은 기능만 쓰자 인것 같다.

그의 말을 인용해보자...

언어의 좋은 점만을 사용하고 나쁜 점은 사용을 자제함으로써 좀더 좋은 프로그래머가 될 수 있습니다.

결국 요점은,
어떻게 하면 나쁜 점을 피해서 좋은 결과를 얻느냐는 것입니다.

표준을 제정하는 위원회가(ECMA) 언어의 나쁜 점들을 제거하기는 어렵습니다.
위원회가 나쁜 점을 제거하게 되는 경우 기존에 그 나쁜 점을 사용하여 개발된 프로그램들에 영향을 미치기 때문입니다.

- 이 책은, 더글라스가 생각하는 좋은 자바스크립트 사용법에 대해 알려주는 책!
- 하지만 그는 괴팍(?) 하기로 나름 유명하다.
- 그래서 우리가 보기에 이상한 주장을 할 수도 있음.
- 그리고 TS를 사용하면 해결될 문제도 있다고 생각함.
- 그러니 우리는 이 책을 읽으면서,
 - 왜 더글라스는 이렇게 생각했지?
 - 수공이가는가? 그렇지 않다면 왜? 에 대해서 한번 생각해 보면 좋을듯 함!

진짜 시작해봅시다!

챕터1.이름

주장1. 이름은 문자로 시작해서 문자로 끝내라

- _나 \$ 로 시작할 수 있지만 하지마라.
 - _ 는 보통 지역 변수를 표현하기 위해 사용하는데, _ 로 표현해줘야 한다는것 자체가 잘못 짚거다.
 - \$ 는 전처리기나 쓰는거다.
- 일단, _ 로 시작하는 이름은 저도 지금 많이 쓰고 있네요,
 - 저도 private 하거나 임시변수인 경우에 많이 쓰고 있습니다.
 - 작가는 이런 암시적 네이밍 을 모두 피하라고 주장합니다.
 - _ 이 private 이라는건 강제할 수 없고, 팀에 들어온 새 팀원이 히스토리를 파악하지 못할 수 있으니깐요.

대안?

이전 글들을 찾아보니, 더글라스는 이런 방식으로 사용하길 권장합니다.

- https://ui.toast.com/weekly-pick/ko_20200312
- 글을 읽어보니, NHN도 _ 를 사용하지 않는 컨벤션으로 변경했네요.

```
function SomeModule() {  
  const privateProp = 'dont touch this';  
  const publicProp = 'you can touch this';  
  
  doSomethingWithPrivateProp = () => { ... }  
  
  const publicMethod = () => {  
    doSomethingWithPrivateProp();  
  }  
  return {  
    publicProp,  
    publicMethod  
  }  
} //우리한테도 아주 익숙한, 클로저를 사용해서 private하게 격리하기.
```

- 근데 타입스크립트의 경우 `private` 자체를 지원하고, NHN 글에서 볼 수 있듯, ECMA에서도 조만간 `private`이 정규 기능으로 올라온다고 하긴 합니다.
- 근데 js 밖에 못쓰는 상황이고, 최신 문법을 바로 사용할 수 없는 상황이라면 위와 같은 방법으로 하는거 좋아보입니다.

주장2. 생성자를 제외한 모든 변수 시작은 소문자여야 한다. + `new` 쓰지마라.

- `new`를 안쓰면 대문자를 쓸 일조차 없다.
- 클래스를 쓰지 말라는건가?
- 자바스크립트에서 클래스와 함수는 너무 모호하다, 오해의 여지가 있다. 오해를 줄이기 위해 쓰지 말라는 뜻으로 해석됩니다(다음 파트에서 예시를 하나 찾아봤어요)

예약어

- 작가는 예약어를 싫어합니다.(이 말 밖에는...)
- 생각보다 예약어를 싫어하는 개발자들은 많습니다...
- 근데 예약어를 변수명으로 쓸 수 있게 되면, 되려 더 헷갈릴것 같은데 개발자 마음은 알다가도 모르겠네요...

숫자

- 자바스크립트는 `number` 라는 타입으로 숫자를 표현합니다.
- 숫자 타입이 하나라는 점은 오히려 편한 요소입니다.
- 자바스크립트는 `IEEE754` 배정도(double)로 숫자를 표현합니다.
- <https://backstreet-programmer.tistory.com/140>
- 여기 원리가 잘 설명되어있으니, 궁금하면 확인!
- <https://ko.wikipedia.org/wiki/부동소수점>
- 위키에도 좋은 자료가 있음.

- 대부분의 언어에서 IEEE754를 사용하고 있어서, 사실 어느 언어나 발생하는 문제
- 자바스크립트는 UI를 통해 유저 입력값을 바로 받아야 하는 경우가 있어서 더 신경쓰이는듯.
 - 우리 팀의 경우 대부분의 연산을 유효숫자만큼 정수로 정규화해서 계산한 후 다시 내림.
 - 가령 소수 10자리까지 계산해야하면, *10 해서 계산
 - 생각보다 소수 두자리만 넘어가도 오차가 많음 π

```
10.10 * 10.10  
> 102.00999999999999  
(101*101) / 100  
> 102.01
```

월정산같이 단위가 큰 경우 0.01도 상당히 큰 금액...

숫자리터럴

- 자바스크립트는 숫자 리터럴을 제공
- 2진수 8진수 10진수 16진수를 표현할 수 있음.
- 숫자 리터럴은 해당 숫자리터럴이 표현할 수 있는 가장 가까운 값을 가져옴.
- 실수 표현이 아니라면 오차에 대해 걱정할 필요는 없을듯!
- 숫자 외에 `NAN` `Infinity` 라는 특별한 숫자를 나타내는 리터럴도 있음.
- 참고로 `NAN` 은 `isNaN` 함수를 사용하지 않으면 비교할 수 없음. `NAN===NAN` 은 `false`임.
- 개인적으로 `NaN` 이 넘버타입인게 진짜 이상한데... 또 생각해보면 넘버타입이 아니면 문제가 발생할 수 있음(유저가 입력해버린다거나...) `NAN`이라는 타입이 있으면 어떨까 생각했는데 그러면 암시적으로 타입이 바뀌어버려서 좀 그런가 싶기도 하고...

Number

- number타입을 만드는 함수
- Number의 시작이 대문자라 new를 써야할 것 같지만 사실 아니다.
 - new Number 하면 완전 의도하지 않게 동작한다.
 - 더글라스는 이런 부분을 우려함. 대소문자같이 암시적인 방법으로 클래스와 함수를 구분하면 사용자 입장에서는 이를 파악하기가 너무 힘들.
 - 그래서 아예 new 자체를 쓰지 말고, 함수만으로 동작할 수 있게 사용하자는 취지.

자바스크립트가 다룰 수 있는 수의 한계

- Number 안에는 유용한 메소드들이 많이 있음
- Number.EPSILON 은 자바스크립트가 표현할 수 있는 가장 작은 수(연산시 값이 변하는 최소값)
 - 수학에서 되게 유용한 개념. 증명에서 많이 사용하는데 알고리즘을 짤때도 쓰임.
- Number.MAX_SAFE_INTEGER 은 자바스크립트가 표현할 수 있는 최대 정수.
 - 대략 9천조. 이 값을 대략적으로 알고있는것은 중요!

- 비슷하게 MAX_VALUE 가 있음.
 - MAX_VALUE 를 넘어가는 값은 INFINITY 리터럴로 수렴된다.
 - MAX_SAFE_INTEGER 보다 큰 값을 저장할 수 있지만 안전하지 않음
- 안전 의 의미는 정수를 정확하고 올바르게 비교할 수 있음을 이야기 함. (결합법칙, 분배법칙, 부등호, 등호 등이 안전하게 연산되는것.)
- 위의 오차들은 오류를 뱉지도 않음. 발견하기 아주 어려운 버그가 될 수 있음. 그러니 MAX_SAFE_INTEGER 의 값을 대략적으로 인지하고, 해당 값 근처에서 수를 다루게 된다면 별도의 방안을 마련해야 함(big.js 등)
- 더글라스는 3,4,5장에 걸쳐서 이걸 직접만들어보는 과정을 진행했는데... 관심이 있다면 보는것으로!

무엇 타입

- true 혹은 false 라는 값만 가지는 타입. -> True | False
 - 프로그래밍 언어론에서는 이러한 타입을 SUM 타입(서로소 유니온) 이라고 부른다
 - <https://ahnheejong.name/articles/safe-data-modeling-using-disjoint-union-type/>
 - 오늘 발표 주제하고는 관련 없지만, 함수형 진형에서 대세라서 틈새 소개를 해보았습니다.

자바스크립트의 truthy falsy

- `false`, `null`, `undefined`, `""`, `0`, `NaN` 은 falsy한 값이다.
- 더글라스는 truthy, falsy 한 값 대신에 진짜 불 값만 조건문에 대입하길 원한다.
- 이게 맞는 말이긴 한데... 자바스크립트 truthy falsy 적절히 쓰면 코드를 훨씬 깔끔하게 작성할 수 있기도 해서(가령 조건문에서 null check 위해서 `if(!어쩌구)`) 아예 이렇게 쓰기는 힘들지 않을까..
 - falsy 값 때문에 발생했던 버그 중 하나는 `const a = petCount || null`
 - `petCount` 가 0 인거랑 선택 안한거랑 비즈니스 로직상에서 다른거였는데, null로 들어가버림.

총평

- 초반부라 큰 내용은 없었음.
- MAX_VALUE 와 SAFE_MAX_INTEGER 의 차이를 몰랐는데 큰 값을 다룰때는 주의해야 할듯.
- 더글러스는 클래스를 싫어한다.(자바스크립트 개발자중에 좀 있는듯)
- 타입스크립트는 클래스 열심히 쓰는데 관련 자료를 좀 찾아봐야겠다고 느꼈습니다.